5/11-31

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/003,570	10/24/2001	Gilbert Cabillic	TIF-32157	5072

23494 7590 06/16/2005

TEXAS INSTRUMENTS INCORPORATED
P O BOX 655474, M/S 3999
DALLAS, TX 75265

EXAMINER	
STEELMAN, MARY J	
ART UNIT	**PAPER NUMBER**
2191	

DATE MAILED: 06/16/2005

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	**Application No.** 10/003,570	**Applicant(s)** CABILLIC ET AL.
	Examiner Mary J. Steelman	**Art Unit** 2191

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *3/14/2005,4/8/2005*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-15 and 18-24* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-15 and 18-24* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *14 March 2005* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☒ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☒ All b)☐ Some * c)☐ None of:

1.☒ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)

2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.

4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.

5) ☐ Notice of Informal Patent Application (PTO-152)

6) ☒ Other: *Copy ofaccepted drawings*.

HL

DETAILED ACTION

1. Per Applicants request, claims 16,17, and 25-27 are canceled. Claims 1-3, 5, 7-15, 18-24 have been amended. Claims 1-15, and 18-24 are pending.

Priority

2. Acknowledgment is made of applicant's claim for foreign priority based on an application filed in European Patent Convention on 11/29/2000. The certified priority document was received 8 April 2005.

Drawings

3. In view of the Replacement Sheet FIG. 1, the prior drawing objection is hereby withdrawn.

Specification

4. In view of the amendments to the Specification, the prior objections are hereby withdrawn.

Examiner notes that the references on pages 28 and 29 have been deleted. The prior objection is hereby withdrawn. Regarding Applicant's comments on page 28, 3rd paragraph, no Form 1449 has been received.

Claim Rejections - 35 USC § 101

5. In view of the amendments to claim 9, 10, 13-15, & 18-21, and the cancellation of claims 16, 17, 25, and 26, the prior 35 USC 101 rejections are hereby withdrawn.

Claim Rejections - 35 USC § 112

6. In view of the amendments to claims 8, 20, and 21, the prior 35 USC 112 second paragraph rejections are hereby withdrawn.

7. The following is a quotation of the second paragraph of 35 U.S.C. 112:

> The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

8. Claim 24 recites the limitation "...said at least two program code modules..." in line 5. There is insufficient antecedent basis for this limitation in the claim. Claim should recite, "...said at least two program source code modules..." Add 'source' to be consistent with limitation on line 3.

Claim Rejections - 35 USC § 103

9. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

10. Claims 1-15, and 18-24 are rejected under 35 U.S.C. 103(a) as being unpatentable over US Patent 6,571,388 B1 to Venkatraman et al., in view of US Patent 6,230,307 B1 to Davis et al.

Per claims 1, 9, 11,and 22:

A method for generating program source code for translating high level code into instructions for a target processor, the method comprising:

(Venkatraman: Col. 2, lines 53-61, "...includes a set of custom environment tools...enable the development of custom software environment...customized in response to...particular functions performed by the application program (method for generating program source code)..." Venkatraman disclosed at col. 4, lines 54-63, a pre-load analyzer that generates a pre-load class list (generating program source code) , which contains the pre-load classes used in translating code for the target processor. Col. 5, lines 32-33, "The virtual machine then interprets (translates) and executes the referenced byte code.")

-generating program code for translating high level code into instructions for a target processor...

(Venkatraman: Col. 2, lines 53-61, "...includes a set of custom environment tools...enable the development of custom software environment...customized in response to...particular functions performed by the application program (method for generating program source code)..." Venkatraman disclosed at col. 4, lines 54-63, a pre-load analyzer that generates a pre-load class list (generating program source code) , which contains the pre-load classes used in translating code for the target processor.)

-determining a program code characteristic corresponding to said target processor;

(Venkatraman: Col. 3, lines 57-60, "The pre-load analyzer analyzes the application program to determine which classes are needed for it to execute on the target device. The pre-load analyzer bases its analysis on a set of methods which are invoked by the application program.")

-deriving one or more program code modules in accordance with said desired program code characteristic;

(Venkatraman: Col. 2, lines 16-17, "...by customizing the procedures for class loading to the available resources in the target device.")

-generating program source code for translating high level code into instructions for said target processor from said one or more program code modules.

(Venkatraman: Col. 2, lines 53-61, "...includes a set of custom environment tools...enable the development of custom software environment...customized in response to...particular functions performed by the application program (method for generating program source code)..." Col. 2, lines 20-23, "The tools build the custom environment including a set of code for a virtual machine and a set of code for each of a set of pre-loaded classed specified..." Venkatraman disclosed at col. 4, lines 54-63, a pre-load analyzer that generates a pre-load class list (generating program source code) , which contains the pre-load classes used in translating code for the target processor (col. 5, lines 32-33).)

Venkatraman disclosed generating program code for translating high level code into instructions for a target processor, which could be an embedded system. However, Davis provided more details regarding (Abstract, lines 1-9) controlling and executing the hardware objects (processors) via high level software constructs and managing the reconfigurable resources..."

Davis disclosed (col. 4, lines 41-43) a method and system for dynamically relocatable hardware object development and execution. Davis disclosed (col. 5, lines 52-61) a source program, compiler / converter into an object program, calls to library routines, resulting in load module as output, to get the executable code into a computer system and executing the program.

Therefore, it would have been obvious, to one of ordinary skill in the art, at the time of the invention to modify Venkatraman's invention to include such disclosures as Davis' for reconfigurable resources, which includes virtual processors, because software used to develop processor is known in the art and is useful for developing and debugging code to be applied to resources.

Per claims 2, 10, and 12:

-generating program source code for translating high level code into instructions for one of a plurality of target processors.

(Venkatraman disclosed generating program source code at col. 4, lines 54-63, a pre-analyzer writes a pre-load class list containing pre-load classes which is used as a source input to the virtual machine (col. 5, lines 32-33). Col. 2, lines 58-61, "The custom software environment is customized in response to the available resources in the target device and the particular functions performed by the application program", col. 3, lines 51-53, "...these techniques are readily adaptable to customize the software environment to multiple application programs in the target device.")

Venkatraman disclosed generating program code for translating high level code into instructions for a target processor, which could be an embedded system. However, Davis provided more details regarding a plurality of target processors (col. 6, lines 40-42), "Either of these embodiments may or may not have an embedded processor or a plurality of embedded processors."

Therefore, it would have been obvious, to one of ordinary skill in the art, at the time of the invention, to modify Venkatraman's invention to include a plurality of processors, whether virtual as in a virtual machine, or hardware, a detected resource, because either situation is well known in the art. A plurality of processors is useful in handling multi-threaded application process more quickly.

Per claim 3:

-forming agglomerated program source code from a plurality of program code modules in accordance with said desired program code characteristic.

(Venkatraman disclosed generating program source code at col. 4, lines 54-63, a pre-analyzer writes a pre-load class list (forming agglomerated program source code) containing pre-load classes which is used as a source input to the virtual machine (col. 5, lines 32-33). Col. 3, lines 51-53, ""'...these techniques are readily adaptable to customize (desired program code characteristic) the software environment to multiple application programs in the target device.")

Per claim 4:

-deriving said program code modules in accordance with a desired functionality for said target processor.

(Venkatraman: As an example, col. 3, lines 26-28, "...specify class libraries from differing sources depending upon the available resources and nature of the target device.")

Per claim 5:

-determining comprises determining respective program code characteristics for respective ones of a plurality of target processors;

(Venkatraman: Col. 3, lines 57-60, "The pre-load analyzer analyzes the application program to determine which classes are needed for it to execute on the target device. The pre-load analyzer bases its analysis on a set of methods which are invoked by the application program.")

-deriving comprises deriving respective program code modules in accordance with said respective program code characteristics;

(Venkatraman: Col. 3, lines 64-66, "The pre-load analyzer takes as input a virtual machine class list. The virtual machine class list specifies a set of native classes which are implemented in the virtual machine.")

-generating comprises generating program source code for translating high level code into instructions for said target processors from said program code modules.

(Venkatraman disclosed generating program source code at col. 4, lines 54-63, a pre-analyzer writes a pre-load class list containing pre-load classes which is used as a source input (generating program source code) to the virtual machine (col. 5, lines 32-33). Venkatraman: Abstract, lines 9-14, "The tools build the custom environment including a set of code for a virtual machine and a set of code for each of a set pre-loaded classes specified in the pre-load class list. The virtual machine loads classes from the pre-loaded classes, a file system class library, and/or a networked class library using widely available network protocols.")

Per claim 6:

-deriving comprises selecting one or more pre-defined program code modules in accordance with said program code characteristic from a plurality of available program code modules.

(Venkatraman: As an example, col. 3, lines 23-28, "...the custom software environment is customized to a particular application of the target device in that the class loader in the virtual machine enables a developer to specify class libraries from differing sources depending upon the available resources and nature of the target device.")

Per claims 7, 18, and 19:

-program code provides a virtual machine for said target processor.

(Venkatraman: Col. 5, lines 39-43, "...application program is a JAVA application and the virtual machine in the custom software environment is a JAVA virtual machine with its native classes implemented in native code for the hardware platform.")

Per claims 8, 20, and 21:

-program code comprises elements of a programming langage.

(Venkatraman: Col. 5, lines 39-43, "...JAVA application ...JAVA virtual machine...")

Per claim 13:

An apparatus, comprising at least one program source code module of a plurality of program source code modules for translating between high level code and instructions for a target processor…

(Venkatraman, col. 2, lines 53-61, "…software development system includes a set of custom environment tools…customized to support…program which executes on a target device…" Venkatraman disclosed generating program source code at col. 4, lines 54-63, a pre-analyzer writes a pre-load class list containing pre-load classes (plurality of program source code modules) which is used as a source input (generating program source code) to the virtual machine (col. 5, lines 32-33) (for translating between high level code and instructions for a target processor)…")

-said at least one program code module corresponding to a characteristic of said target processor and being selected from said plurality of program source code modules.

(Venkatraman disclosed at col. 3, lines 26-28 selecting from class libraries depending on the nature of the target device.)

Per claim 14:

-at least one additional program code modules for translating between high level code and instructions for respective ones of at least two target processors.

(Venkatraman: Col. 2, lines 58-65, "The custom software environment is customized in response to the available resources in the target device…assemble together only those classes that are needed to support a desired application …")

Per claim 15:

-said at least two program code modules are selected from a plurality of predefined program code modules.

(Venkatraman: Col. 2, lines 23-28, "The virtual machine loads classes from the pre-loaded classes, a file system class library, and/or a networked class library…")

Per claim 23:

A processor, configured by program code comprising an agglomeration of two or more program source code modules of said plurality of said program source code modules.

(Venkatraman: Col. 3, lines 23-28, "…custom software environment is customized to a particular application of the target device in that the class loader in the virtual machine enables a developer to specify class libraries from differing sources…" Additionally, see rejection of claims 1 and 3 above.)

Per clam 24:

-A system comprising a first and a second processor, said first and second processor configured in accordance with program code comprising at least two program source code modules, wherein the first of said at least two program source code modules is arranged to translate high level code

to instructions for said first processor and a second of said at least two program code modules is arranged to translate high level code to instructions for said second processor.
(Venkatraman disclosed generating program source code at col. 4, lines 54-63, a pre-analyzer writes a pre-load class list containing pre-load classes which is used as a source input (generating program source code) to the virtual machine (col. 5, lines 32-33). Venkatraman: Col. 2, lines 55-61, "The custom environment tools enable the development of a custom software environment which executes on a target device. The custom software environment is customized in response to the available resources in the target device and the particular functions performed by the application program." Additionally, see rejection of claims 13 and 14 above.)

Response to Arguments

11. Applicant's arguments received 8 April 2005 have been fully considered but they are not persuasive.

Applicant has amended claims to recite "generating/ creating/ forming program source code." Examiner has shown in rejection above that Venkatraman disclosed generating program source code at col. 4, lines 54-63, a pre-analyzer writes a pre-load class list containing pre-load classes which is used as a source input (generating program source code) to the virtual machine (col. 5, lines 32-33). See rejection of limitations above. Examiner maintains rejection of claims 1-15, and 18-24.

Conclusion

12. Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

13. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Mary Steelman, whose telephone number is (571) 272-3704. The examiner can normally be reached Monday through Thursday, from 7:00 AM to 5:30 PM If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Tuan Q. Dam can be reached at (571) 272-3695. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Any inquiry of a general nature or relating to the status of this application should be directed to the TC 2100 Group receptionist: 571-272-2100.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Mary Steelman
06/01/2005

WEI Y. ZHEN
PRIMARY EXAMINER

Accepted by MS 6-5-05

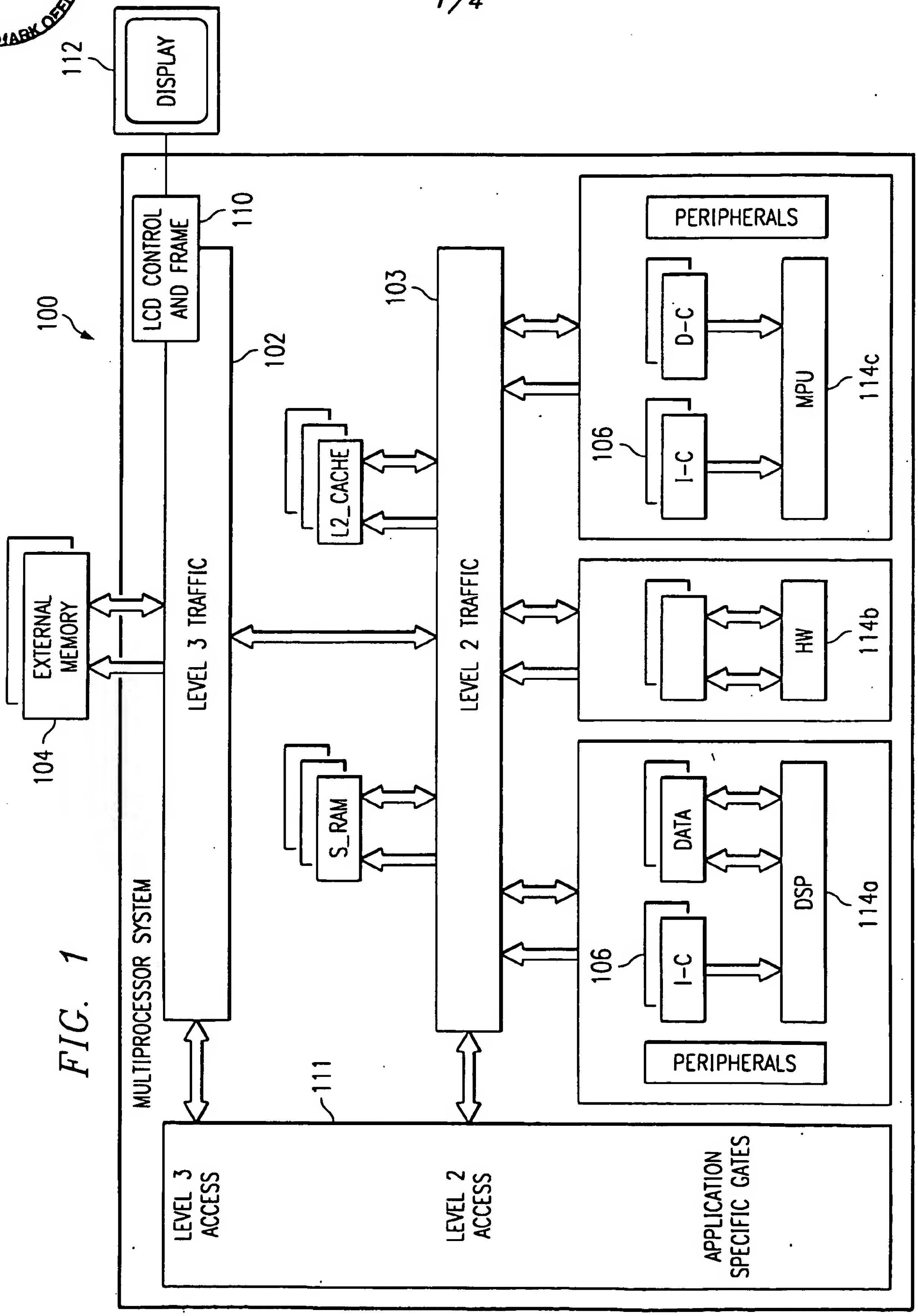

FIG. 1

Accepted by Examiner 6.5.05

FIG. 2

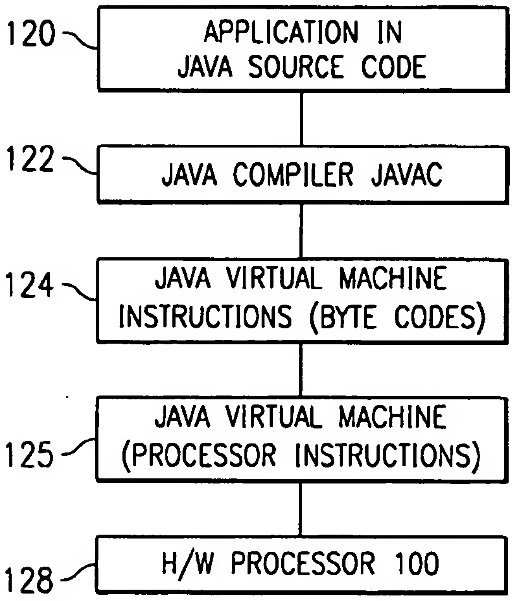
120
APPLICATION IN JAVA SOURCE CODE
122
JAVA COMPILER JAVAC
124
JAVA VIRTUAL MACHINE INSTRUCTIONS (BYTE CODES)
125
JAVA VIRTUAL MACHINE (PROCESSOR INSTRUCTIONS)
128
H/W PROCESSOR 100

FIG. 5

INTEL PENTIUM II				TI DSP C55X			
	LANGUAGE	CYCLE	MEMORY		LANGUAGE	CYCLE	MEMORY
SWITCH THREADED RATIO	ANSIC GNUC	162 132 19%	20 80 400%	SWITCH THREADED RATIO	ANSIC C/ANSI	294 183 62%	40 80 200%
				SWITCH SWITCH-2 RATIO	ANSIC ANSIC	294 622 211%	40 20 50%

Accepted by Examiner 6.5.05 MS

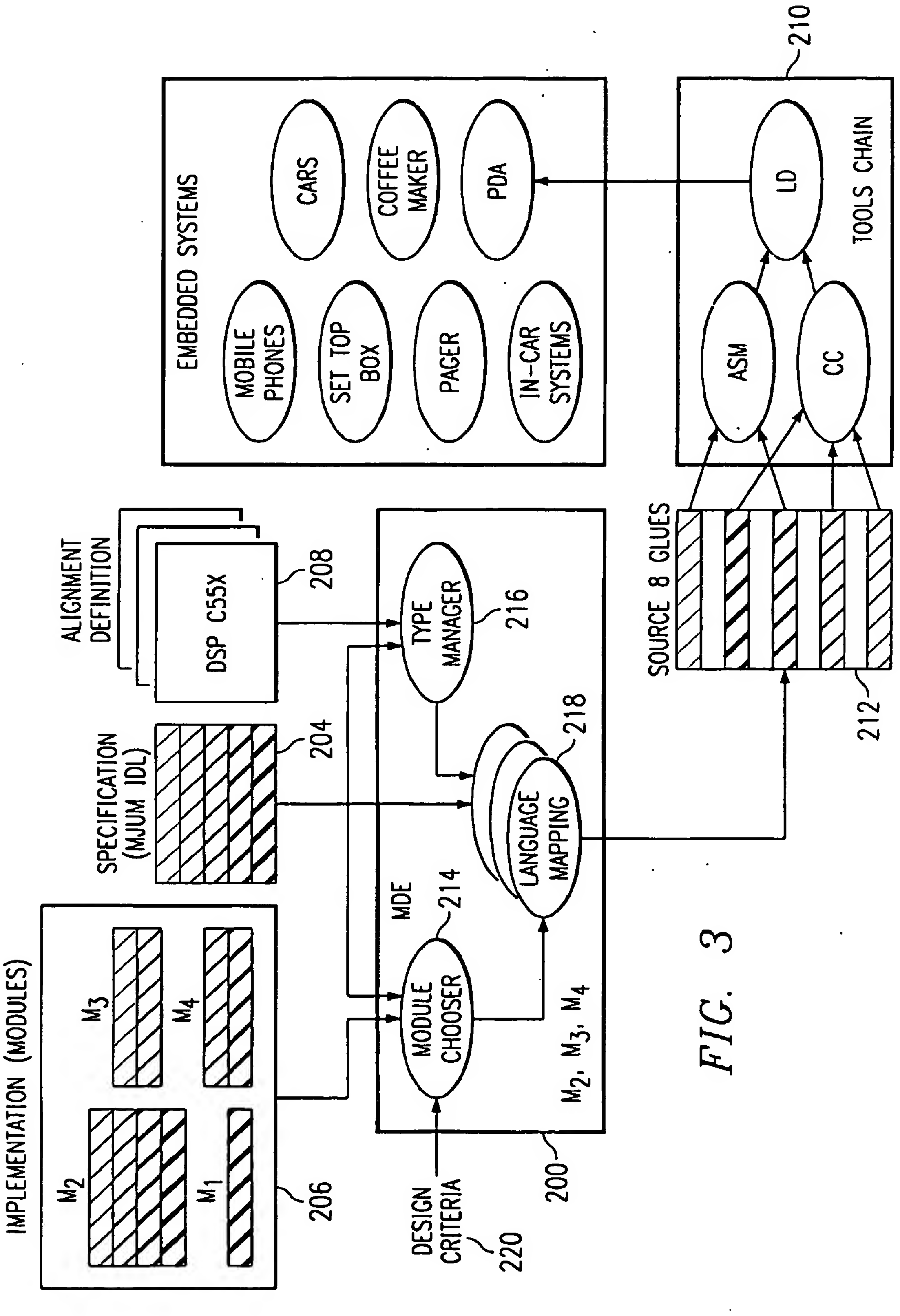

FIG. 3

Accepted by Examiner 6.5.05 MK

FIG. 4

MODULE	DESCRIPTION
	ARITHMETIC
INTEGER	REPRESENTATION, OPERATIONS AND CONVERSION OF JAVA INT
LONG	REPRESENTATION, OPERATIONS AND CONVERSION OF JAVA LONG
FLOAT	REPRESENTATION, OPERATIONS AND CONVERSION OF JAVA FLOAT
DOUBLE	REPRESENTATION, OPERATIONS AND CONVERSION OF JAVA DOUBLE
	JAVA FRAME
LOCAL	LOCAL VARIABLES MANAGEMENT AND ACCESS
STACK	STACK MANAGEMENT AND ACCESS
	CONTROL FLOW
MOTOR	INTERPRETATION ENGINE
BRANCH	BRANCHING AND SUBROUTINE
EXCEPTION	EXCEPTION MANAGEMENT AND BRANCHING
	OBJECT
OBJECT	REPRESENTATION AND OPERATIONS ON OBJECT AND REFERENCE
FIELD	REPRESENTAION OF 'SMALL' TYPE AND FIELD ACCESS
ARRAY	REPRESENTATION AND OPERATIONS ON ARRAY AND ELEMENT
METHOD	METHOD INVOCATION AND PARAMETERS PASSING
INTERFACE	INTERFACE MANAGEMENT
STRING	INTERNAL STRING AND JAVA STRING REPRESENTATION
	CLASS
CIREPOS	CLASS REPOSITORY
LOAD	CLASS LOADER MANAGER
	MISCELLANEOUS
JNI	JAVA NATIVE INTERFACE SUPPORT
THREAD	THREAD AND MONITOR SUPPORT